IPPM H. Song
Internet-Draft Futurewei
Intended status: Standards Track B. Gafni
Expires: December 17, 2022 Nvidia
F. Brockners
Cisco
S. Bhandari
Thoughtspot
T. Mizrahi
Huawei
June 15, 2022

In-situ OAM Direct Exporting
draft-ietf-ippm-ioam-direct-export-09

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) is used for recording and collecting operational and telemetry information. Specifically, IOAM allows telemetry data to be pushed into data packets while they traverse the network. This document introduces a new IOAM option type (denoted IOAM-Option-Type) called the Direct Export (DEX) Option-Type, which is used as a trigger for IOAM data to be directly exported or locally aggregated without being pushed into in-flight data packets. The exporting method and format are outside the scope of this document.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

IOAM [RFC9197] is used for monitoring traffic in the network, and for incorporating IOAM data fields (denoted IOAM-Data-Fields) into in-flight data packets.

IOAM makes use of four possible IOAM-Option-Types, defined in [RFC9197]: Pre-allocated Trace Option-Type, Incremental Trace Option-

Type, Proof of Transit (POT) Option-Type, and Edge-to-Edge Option-Type.

This document defines a new IOAM-Option-Type called the Direct Export (DEX) Option-Type. This Option-Type is used as a trigger for IOAM nodes to locally aggregate and process IOAM data, and/or to export it to a receiving entity (or entities). Throughout the document this functionality is referred to as collection and/or exporting. A "receiving entity" in this context can be, for example, an external collector, analyzer, controller, decapsulating node, or a software module in one of the IOAM nodes.

Note that even though the IOAM-Option-Type is called "Direct Export", it depends on the deployment whether the receipt of a packet with DEX Option-Type leads to the creation of another packet. Some deployments might simply use the packet with the DEX Option-Type to trigger local processing of OAM data. The functionality of this local processing is not within the scope of this document.

This draft has evolved from combining some of the concepts of PBT-I from [I-D.song-ippm-postcard-based-telemetry] with immediate exporting from [I-D.ietf-ippm-ioam-flags].

2. Conventions

2.1. Requirement Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Terminology

Abbreviations used in this document:

IOAM: In-situ Operations, Administration, and Maintenance

OAM: Operations, Administration, and Maintenance [RFC6291]

DEX: Direct EXporting

3. The Direct Exporting (DEX) IOAM-Option-Type

3.1. Overview

The DEX Option-Type is used as a trigger for collecting IOAM data locally or for exporting it to a receiving entity (or entities). Specifically, the DEX Option-Type can be used as a trigger for collecting IOAM data by an IOAM node and locally aggregating it; thus, this aggregated data can be periodically pushed to a receiving entity, or pulled by a receiving entity on-demand.

This Option-Type is incorporated into data packets by an IOAM encapsulating node, and removed by an IOAM decapsulating node, as illustrated in Figure 1. The Option-Type can be read but not modified by transit nodes. Note: the terms IOAM encapsulating, decapsulating and transit nodes are as defined in [RFC9197].

```
                                      ^
                                      |Exported IOAM data
                                      |
                                      |
                                      |
                +--------------+------+-------+--------------+
                |              |              |              |
                |              |              |              |
  User      +---+----+     +---+----+     +---+----+     +---+----+
  packets   |Encapsu-|     | Transit|     | Transit|     |Decapsu-|
  --------->|lating  |====>| Node   |====>| Node   |====>|lating  |---->
            |Node    |     | A      |     | B      |     |Node    |
            +--------+     +--------+     +--------+     +--------+
```

```
        Insert DEX       Export         Export        Remove DEX
        option and      IOAM data      IOAM data      option and
        export data                                   export data
```

Figure 1: DEX Architecture

The DEX Option-Type is used as a trigger to collect and/or export IOAM data. The trigger applies to transit nodes, the decapsulating node, and the encapsulating node:

- An IOAM encapsulating node configured to incorporate the DEX Option-Type encapsulates (possibly a subset of) the packets it forwards with the DEX Option-Type, and MAY export and/or collect the requested IOAM data immediately. Only IOAM encapsulating nodes are allowed to add the DEX Option-Type to a packet. An IOAM encapsulating node can generate probe packets that incorporate the DEX Option-Type. These probe packets can be generated periodically or on-demand (for example triggered by the management

  plane). The specification of such probe packets is outside the scope of this document.

- A transit node that processes a packet with the DEX Option-Type MAY export and/or collect the requested IOAM data.

- An IOAM decapsulating node that processes a packet with the DEX Option-Type MAY export and/or collect the requested IOAM data, and MUST decapsulate the IOAM header.

As in [RFC9197], the DEX Option-Type can be incorporated into all or a subset of the traffic that is forwarded by the encapsulating node, as further discussed in Section 3.1.1 below. Moreover, IOAM nodes respond to the DEX trigger by exporting and/or collecting IOAM data either for all traversing packets that carry the DEX Option-Type, or selectively only for a subset of these packets, as further discussed in Section 3.1.2 below.

3.1.1. DEX Packet Selection

If an IOAM encapsulating node incorporates the DEX Option-Type into all the traffic it forwards it may lead to an excessive amount of exported data, which may overload the network and the receiving entity. Therefore, an IOAM encapsulating node that supports the DEX Option-Type MUST support the ability to incorporate the DEX Option-Type selectively into a subset of the packets that are forwarded by it.

Various methods of packet selection and sampling have been previously defined, such as [RFC7014] and [RFC5475]. Similar techniques can be applied by an IOAM encapsulating node to apply DEX to a subset of the forwarded traffic.

The subset of traffic that is forwarded or transmitted with a DEX Option-Type SHOULD NOT exceed 1/N of the interface capacity on any of the IOAM encapsulating node's interfaces. It is noted that this requirement applies to the total traffic that incorporates a DEX Option-Type, including traffic that is forwarded by the IOAM encapsulating node and probe packets that are generated by the IOAM encapsulating node. In this context N is a parameter that can be configurable by network operators. If there is an upper bound, M, on the number of IOAM transit nodes in any path in the network, then it is recommended to use an N such that N >> M. The rationale is that a packet that includes a DEX Option-Type may trigger an exported packet from each IOAM transit node along the path for a total of M exported packets. Thus, if N >> M then the number of exported packets is significantly lower than the number of data packets forwarded by the

IOAM encapsulating node. If there is no prior knowledge about the network topology or size, it is recommended to use N>100.

3.1.2. Responding to the DEX Trigger

The DEX Option-Type specifies which IOAM-Data-Fields should be exported and/or collected, as specified in Section 3.2. As mentioned above, the data can be locally collected, and optionally can be aggregated and exported to a receiving entity, either proactively or on-demand. If IOAM data is exported, the format and encapsulation of the packet that contains the exported data is not within the scope of the current document. For example, the export format can be based on [I-D.spiegel-ippm-ioam-rawexport].

An IOAM node that performs DEX-triggered exporting MUST support the ability to limit the rate of the exported packets. The rate of exported packets SHOULD be limited so that the number of exported packets is significantly lower than the number of packets that are forwarded by the device. The exported data rate SHOULD NOT exceed 1/N of the interface capacity on any of the IOAM node's interfaces. It is recommended to use N>100. Depending on the IOAM node's architecture considerations, the export rate may be limited to a lower number in order to avoid loading the IOAM node. An IOAM node MAY maintain a counter or a set of counters that count the events in which the IOAM node receives a packet with the DEX Option-Type and does not collect and/or export data due to the rate limits.

Exported packets SHOULD NOT be exported over a path or a tunnel that is subject to IOAM direct exporting. Furthermore, IOAM encapsulating nodes that can identify a packet as an IOAM exported packet MUST NOT push a DEX Option-Type into such a packet. This requirement is intended to prevent nested exporting and/or exporting loops.

A transit or decapsulating IOAM node that receives an unknown IOAM-Option-Type ignores it (as defined in [RFC9197]), and specifically nodes that do not support the DEX Option-Type ignore it. Note that as per [RFC9197] a decapsulating node removes the IOAM encapsulation and all its IOAM-Option-Types, and specifically in the case where one of these options is a (possibly unknown) DEX Option-Type. The ability to skip over a (possibly unknown) DEX Option-Type in the parsing or in the decapsulation procedure is dependent on the specific encapsulation, which is outside the scope of this document. For example, when IOAM is encapsulated in IPv6 [I-D.ietf-ippm-ioam-ipv6-options] the DEX Option-Type is incorporated either in a Hop-by-Hop options header or in a Destination options header, and thus can be skipped using the length field in the options header.

3.2. The DEX Option-Type Format

The format of the DEX Option-Type is depicted in Figure 2. The length of the DEX Option-Type is at least 8 octets. The DEX Option-Type MAY include one or more optional fields. The existence of the optional fields is indicated by the corresponding flags in the Extension-Flags field. Two optional fields are defined in this document, the Flow ID and the Sequence Number fields. Every optional field MUST be exactly 4 octets long. Thus, the Extension-Flags field explicitly indicates the length of the DEX Option-Type. Defining a new optional field requires an allocation of a corresponding flag in the Extension-Flags field, as specified in Section 4.2.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Namespace-ID           |     Flags     |Extension-Flags|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               IOAM-Trace-Type                 |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Flow ID (optional)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Sequence Number  (Optional)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: DEX Option-Type Format

Namespace-ID   A 16-bit identifier of the IOAM namespace, as defined in [RFC9197].

Flags   An 8-bit field, comprised of 8 one-bit subfields. Flags are allocated by IANA, as defined in Section 4.2.

Extension-Flags An 8-bit field, comprised of 8 one-bit subfields. Extension-Flags are allocated by IANA, as defined in Section 4.3. Every bit in the Extension-Flag field that is set to 1 indicates the existence of a corresponding optional 4-octet field. An IOAM node that receives a DEX Option-Type with an unknown flag set to 1 MUST ignore the corresponding optional field.

IOAM-Trace-Type A 24-bit identifier which specifies which IOAM-Data-Fields should be exported. The format of this field is as defined in [RFC9197]. Specifically, the bit

that corresponds to the Checksum Complement IOAM-

Data-Field SHOULD be assigned to be zero by the IOAM encapsulating node, and ignored by transit and decapsulating nodes. The reason for this is that the Checksum Complement is intended for in-flight packet modifications and is not relevant for direct exporting.

Reserved: This field SHOULD be ignored by the receiver.

Optional fields: The optional fields, if present, reside after the Reserved field. The order of the optional fields is according to the respective bits that are enabled in the Extension-Flags field. Each optional field is 4 octets long.

Flow ID: An optional 32-bit field representing the flow identifier. If the actual Flow ID is shorter than 32 bits, it is zero padded in its most significant bits. The field is set at the encapsulating node. The Flow ID can be used to correlate the exported data of the same flow from multiple nodes and from multiple packets. Flow ID values are expected to be allocated in a way that avoids collisions. For example, random assignment of Flow ID values can be subject to birthday problem conflicts, while centralized allocation can avoid this problem. The specification of the Flow ID allocation method is not within the scope of this document.

Sequence Number: An optional 32-bit sequence number starting from 0 and increasing by 1 for each following monitored packet from the same flow at the encapsulating node. The Sequence Number, when combined with the Flow ID, provides a convenient approach to correlate the exported data from the same user packet.

4. IANA Considerations

4.1. IOAM Type

The "IOAM Type Registry" was defined in Section 7.2 of [RFC9197]. IANA is requested to allocate the following code point from the "IOAM Type Registry" as follows:

TBD-type   IOAM Direct Export (DEX) Option-Type

If possible, IANA is requested to allocate code point 4 (TBD-type).

4.2. IOAM DEX Flags

IANA is requested to define an "IOAM DEX Flags" registry. This registry includes 8 flag bits. Allocation is based on the "RFC Required" procedure, as defined in [RFC8126].

New registration requests MUST use the following template:

Bit: Desired bit to be allocated in the 8 bit Flags field of the DEX Option-Type.

Description: Brief description of the newly registered bit.

Reference: Reference to the document that defines the new bit.

4.3. IOAM DEX Extension-Flags

IANA is requested to define an "IOAM DEX Extension-Flags" registry. This registry includes 8 flag bits. Bit 0 (the most significant bit) and bit 1 in the registry are allocated by this document, and described in Section 3.2. Allocation of the other bits should be performed based on the "RFC Required" procedure, as defined in [RFC8126].

Bit 0 "Flow ID [RFC XXXX] [RFC Editor: please replace with the RFC number of the current document]"

Bit 1 "Sequence Number [RFC XXXX] [RFC Editor: please replace with the RFC number of the current document]"

New registration requests MUST use the following template:

Bit: Desired bit to be allocated in the 8 bit Extension-Flags field of the DEX Option-Type.

Description: Brief description of the newly registered bit.

Reference: Reference to the document that defines the new bit.

5. Performance Considerations

The DEX Option-Type triggers IOAM data to be collected and/or exported packets to be exported to a receiving entity (or entities). In some cases this may impact the receiving entity's performance, or the performance along the paths leading to it.

Therefore, the performance impact of these exported packets is limited by taking two measures: at the encapsulating nodes, by

selective DEX encapsulation (Section 3.1.1), and at the transit nodes, by limiting exporting rate (Section 3.1.2). These two measures ensure that direct exporting is used at a rate that does not significantly affect the network bandwidth, and does not overload the receiving entity. Moreover, it is possible to load balance the exported data among multiple receiving entities, although the exporting method is not within the scope of this document.

It should be noted that in some networks DEX data may be exported over an out-of-band network, in which a large volume of exported traffic does not compromise user traffic. In this case an operator may choose to disable the exporting rate limiting.

6. Security Considerations

The security considerations of IOAM in general are discussed in [RFC9197]. Specifically, an attacker may try to use the functionality that is defined in this document to attack the network.

An attacker may attempt to overload network devices by injecting synthetic packets that include the DEX Option-Type. Similarly, an on-path attacker may maliciously incorporate the DEX Option-Type into transit packets, or maliciously remove it from packets in which it is incorporated.

Forcing DEX, either in synthetic packets or in transit packets may overload the IOAM nodes and/or the receiving entity (or entities). Since this mechanism affects multiple devices along the network path, it potentially amplifies the effect on the network bandwidth, on the storage of the devices that collect the data, and on the receiving entity's load.

The amplification effect of DEX may be worse in wide area networks in which there are multiple IOAM domains. For example, if DEX is used in IOAM domain 1 for exporting IOAM data to a receiving entity, then the exported packets of domain 1 can be forwarded through IOAM domain 2, in which they are subject to DEX. The exported packets of domain 2 may in turn be forwarded through another IOAM domain (or through domain 1), and theoretically this recursive amplification may continue infinitely.

In order to mitigate the attacks described above, the following requirements (Section 3) have been defined:

- o Selective DEX (Section 3.1.1) is applied by IOAM encapsulating nodes in order to limit the potential impact of DEX attacks to a small fraction of the traffic.

- o Rate limiting of exported traffic (Section 3.1.2) is applied by IOAM nodes in order to prevent overloading attacks and in order to significantly limit the scale of amplification attacks.

- o IOAM encapsulating nodes are required to avoid pushing the DEX Option-Type into IOAM exported packets (Section 3.1.2), thus preventing some of the amplification and export loop scenarios.

Although the exporting method is not within the scope of this document, any exporting method MUST secure the exported data from the IOAM node to the receiving entity. Specifically, an IOAM node that performs DEX exporting MUST send the exported data to a pre-configured trusted receiving entity. Furthermore, an IOAM node MUST gain explicit consent to export data to a receiving entity before starting to send exported data.

An attacker may keep track of the information sent in DEX headers as a means of reconnaissance. This form of recon can be mitigated to some extent by careful allocation of the Flow ID and Sequence Number space, in a way that does not compromise privacy aspects such as customer identities.

The integrity of the DEX Option-Type can be protected through a mechanism of the encapsulating protocol. While [I-D.ietf-ippm-ioam-data-integrity] introduces an integrity protection mechanism that protects the integrity of IOAM-Data-Fields, the DEX Option-Type does not include IOAM-Data-Fields, and therefore these integrity protection mechanisms are not applicable to the DEX Option-Type. As discussed in the threat analysis of [I-D.ietf-ippm-ioam-data-integrity], injection or modification of IOAM-Option-Type headers are threats that are not addressed in IOAM.

IOAM is assumed to be deployed in a restricted administrative domain, thus limiting the scope of the threats above and their affect. This is a fundamental assumption with respect to the security aspects of IOAM, as further discussed in [RFC9197].

7. Acknowledgments

The authors thank Martin Duke, Tommy Pauly, Meral Shirazipour, Colin Perkins, Stephen Farrell, Linda Dunbar, Justin Iurman, Greg Mirsky, and other members of the IPPM working group for many helpful comments.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5475] Zseby, T., Molina, M., Duffield, N., Niccolini, S., and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection", RFC 5475, DOI 10.17487/RFC5475, March 2009, <https://www.rfc-editor.org/info/rfc5475>.

[RFC7014] D'Antonio, S., Zseby, T., Henke, C., and L. Peluso, "Flow Selection Techniques", RFC 7014, DOI 10.17487/RFC7014, September 2013, <https://www.rfc-editor.org/info/rfc7014>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC9197] Brockners, F., Ed., Bhandari, S., Ed., and T. Mizrahi, Ed., "Data Fields for In Situ Operations, Administration, and Maintenance (IOAM)", RFC 9197, DOI 10.17487/RFC9197, May 2022, <https://www.rfc-editor.org/info/rfc9197>.

8.2. Informative References

[I-D.ietf-ippm-ioam-data-integrity]
Brockners, F., Bhandari, S., Mizrahi, T., and J. Iurman, "Integrity of In-situ OAM Data Fields", draft-ietf-ippm-ioam-data-integrity-01 (work in progress), March 2022.

[I-D.ietf-ippm-ioam-flags]
Mizrahi, T., Brockners, F., Bhandari, S., Gafni, B., and M. Spiegel, "In-situ OAM Loopback and Active Flags", draft-ietf-ippm-ioam-flags-08 (work in progress), May 2022.

[I-D.ietf-ippm-ioam-ipv6-options]
Bhandari, S. and F. Brockners, "In-situ OAM IPv6 Options", draft-ietf-ippm-ioam-ipv6-options-07 (work in progress), February 2022.

[I-D.song-ippm-postcard-based-telemetry]
Song, H., Mirsky, G., Filsfils, C., Abdelsalam, A., Zhou, T., Li, Z., Mishra, G., Shin, J., and K. Lee, "In-Situ OAM Marking-based Direct Export", draft-song-ippm-postcard-based-telemetry-12 (work in progress), May 2022.

[I-D.spiegel-ippm-ioam-rawexport]
Spiegel, M., Brockners, F., Bhandari, S., and R. Sivakolundu, "In-situ OAM raw data export with IPFIX", draft-spiegel-ippm-ioam-rawexport-06 (work in progress), February 2022.

[RFC6291] Andersson, L., van Helvoort, H., Bonica, R., Romascanu, D., and S. Mansfield, "Guidelines for the Use of the "OAM" Acronym in the IETF", BCP 161, RFC 6291, DOI 10.17487/RFC6291, June 2011, <https://www.rfc-editor.org/info/rfc6291>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

Appendix A. Hop Limit in Direct Exporting

In order to help correlate and order the exported packets, it is possible to include the Hop_Lim/Node_ID IOAM-Data-Field in exported packets; if the IOAM-Trace-Type [RFC9197] has the Hop_Lim/Node_ID bit set, then exported packets include the Hop_Lim/Node_ID IOAM-Data-Field, which contains the TTL/Hop Limit value from a lower layer protocol.

An alternative approach was considered during the design of this document, according to which a 1-octet Hop Count field would be included in the DEX header (presumably by claiming some space from the Flags field). The Hop Limit would starts from 0 at the encapsulating node and be incremented by each IOAM transit node that supports the DEX Option-Type. In this approach the Hop Count field value would also be included in the exported packet.

Contributors

The Editors would like to recognize the contributions of the following individuals to this document.

Tianran Zhou
Huawei
156 Beiqing Rd.
Beijing 100095
China

Email: zhoutianran@huawei.com

Zhenbin Li
Huawei

156 Beiqing Rd.
Beijing 100095
China

Email: lizhenbin@huawei.com

Ramesh Sivakolundu
Cisco Systems, Inc.
170 West Tasman Dr.
SAN JOSE, CA 95134
U.S.A.

Email: sramesh@cisco.com

Authors' Addresses

Haoyu Song
Futurewei
2330 Central Expressway
Santa Clara 95050
USA

Email: haoyu.song@futurewei.com

Barak Gafni
Nvidia
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.

Email: gbarak@nvidia.com

Frank Brockners
Cisco Systems, Inc.
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549

Germany

Email: fbrockne@cisco.com

Shwetha Bhandari
Thoughtspot
3rd Floor, Indiqube Orion, 24th Main Rd, Garden Layout, HSR Layout
Bangalore, KARNATAKA 560 102
India

Email: shwetha.bhandari@thoughtspot.com

Tal Mizrahi
Huawei
8-2 Matam
Haifa 3190501
Israel

Email: tal.mizrahi.phd@gmail.com